

REGULAR EXPRESSIONS (Düzenli İfadeler)

▼ Nedir?

- Metinsel yapılanmalarda belirli koşulları sağlayabilen ifadelerdir.
- Metinsel ifade içerisinde ihtiyaca istinaden düzenlenirler.
- Örnek:

◆ Bir metinsel ifade içerisinde @ karakteri geçen bütün aralıkları elde etmek istediğimizi varsayalım.

`baasdseda@asm1231232'dkasd@dlqwödq213'asdasdwd`

◆ Dikkat ederseniz bu değerlerdeki karakterlerin uzunluğu ve ne olduğu önemli değil. Yeter ki @ karakteri olsun.

▼ Regex

◆ Bu yüzden C#'ta bu tür düzenli ifadeleri temsil edebilmek için Regular Expressions operatörleri geliştirilmiştir.

◆ Bu operatörler eşliğinde elde edilen verinin tasarlanan metinsel düzene uyup uymadığı değerlendirilebilmektedir.

◆ Regular ifadeler System.Text.RegularExpressions namespace'i altındaki Regex sınıfı tarafından temsil edilmektedir.

▼ Regular Expressions Operatörleri

▼ ^ operatörü

- Satır başında istenilen ifadeyle başlanmasını sağlar

```

static void Main(string[] args)
{
    #region Regular Expressions Operators
    //^ Operatörü : Satır başının istenilen ifadeyle başlamasını sağlar.

    //^9 : 9asdasdj, 9123123, 9''!3123, aksdmi 123, 1239, 0912312

    string text = "912367123aklsdmkasdkansdkasd";
    Regex regex = new Regex("^91");
    Match match = regex.Match(text);

    Console.WriteLine(match.Success);

    #endregion
}

```

Microsoft Visual Studio Debug Console

True

C:\Users\Gençay\Desktop\duzenli_ifadele
ss 50844) exited with code 0.
To automatically close the console when
e the console when debugging stops.
Press any key to close this window . .

▼ \ operatörü

```

0 references
class Program
{
    0 references
    static void Main(string[] args)
    {
        #region Regular Expressions Operators
        ^ Operatörü
        #region \ Operatörü
        //\ : Belirli karakter gruplarını içermesini istiyorsak kullanırız.

        //\D : Metinsel değerin ilgili yerinde rakam olmayan tek bir karakterin bulunması gerektiği belirtilir.
        //\d : Metinsel değerin ilgili yerinde 0 - 9 arasında tek bir sayı olacağı ifade edilir.

        //\W : Metinsel değerin ilgili yerinde alfanümerik olmayan karakterin olması gerektiği belirtilir. Alfanümerik
        karakterler : a-z A-Z 0-9
        //\w : Metinsel değerin ilgili yerinde alfanümerik olan karakterin olacağı ifade edilir.

        //\S : Metinsel değerin ilgili yerinde boşluk karakterleri(tab/space) dışında herhangi bir karakterin olabileceği
        belirtilir.
        //\s : Metinsel değerin ilgili yerinde sadece boşluk karakterinin olacağı ifade edilir.
        #endregion
        #endregion
    }
}

```

- Örnek

```
//Örn;
//9 ile başlayan, ikinci karakteri herhangi bir sayı olan ve son karakteri de boşluk olmayan bir düzenli ifade
oluşturalım.
//^9\d\S
#endregion
string text = "9212367123aklsdmkasdkansdkasd";
Regex regex = new Regex(@"^9\d\S");
Match match = regex.Match(text);

Console.WriteLine(match.Success);
#endregion
```

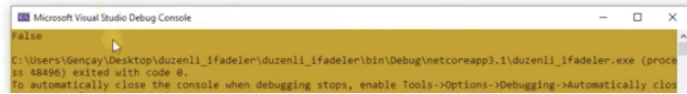


▼ + operatörü

```
#region + Operatörü
//Belirtilen gruptaki karakterlerden bir ya da daha fazlasının olmasını istiyorsak kullanılan operatördür.
//9 ile başlayan, arada herhangi bir sayısal değerleri olan ve son karakteri de boşluk olmayan bir düzenli ifade
oluşturalım.
//^9\d+\S
#endregion
```

```
string text = "98 58345d";
Regex regex = new Regex(@"^9\d+\S");
Match match = regex.Match(text);

Console.WriteLine(match.Success);
#endregion
```



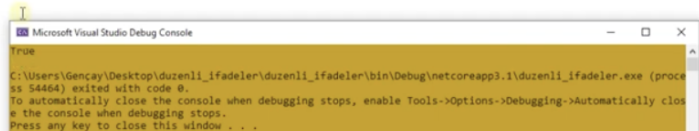
▼ | operatörü

```
#region | Operatörü
//Birden fazla karakter grubundan bir ya da birkaçının ilgili yerde olabileceğini belirtmek istiyorsak mantıksal
veya operatörü kullanılır.

//Baş harfi a ya da b ya da c olan metinsel ifadeyi girelim.
//a|b|c
#endregion
```

```
string text = "ahmet";
Regex regex = new Regex(@"a|b|c");
Match match = regex.Match(text);

Console.WriteLine(match.Success);
#endregion
```



▼ {n} operatörü

```
#region {n} Operatörü
//Sabit sayıda karakterin olması isteniyorsa {adet} şeklinde belirtilmeli.

//507-7514592
//\d\d\d-\d\d\d\d\d\d
//\d{3}-\d{6}
#endregion
```

▼ ? operatörü

```
{n} Operatörü
#region ? Operatörü
//Bu karakterin önüne gelen karakter en fazla bir en az sıfır defa olabilmektedir.

//\d{3}B?A ---> 234BA, 543BA, 543A, 123BBA
#endregion
```

▼ . operatörü

```
#region . Operatörü
//\n : Kullanıldığı yerde \n karakteri dışında herhangi bir karakter bulunabilir.
//\d{3}.A
#endregion
```

```
string text = "123'A";
Regex regex = new Regex(@"\d{3}.A");
```

▼ \b ve \B operatörü

```
. Operatörü
#region \b \B Operatörleri
//\B: Bu ifade ile kelimenin başında ya da sonunda olmaması gereken karakterler bildirilir.
//\b: Bu ifade ilgili kelimenin belirtilen karakter dizisi ile sonlanmasını sağlar.
```

```
//\d{3}dır\B => 123dır, dır123, 123dır2
#endregion
```

```
string text = "123dır";
Regex regex = new Regex(@"\d{3}dır\B");
Match match = regex.Match(text);
```

```
Console.WriteLine(match.Success);
//-----
```

```
Microsoft Visual Studio Debug Console
False
C:\Users\Gençay\Desktop\duzenli_ifadeler\duzenli_ifadeler\bin\Debug\netcoreapp
ss 47996) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options
e the console when debugging stops.
Press any key to close this window . . .
```

▼ [n] operatörü

\b \B Operatörleri

#region [n] Operatörü

//[n] : Karakter aralığı belirtilebilir.

//Ayrıca özel karakterlerin yerinde yazılmasında ifade eder.

//\d{3}[A-E]

//(507) 751 45 92

//[(\d{3})]\s\d{3}\s\d{2}\s\d{2}

#endregion

string text = "(507) 751 45 92";

Regex regex = new Regex(@"[(\d{3})]\s\d{3}\s\d{2}\s\d{2}");

Match match = regex.Match(text);

Microsoft Visual Studio Debug Console

True

C:\Users\Gençay\Desktop\duzenli_ifadele
ss 23396) exited with code 0.
To automatically close the console when
e the console when debugging stops.
Press any key to close this window . .

▼ Regular Expressions Property'leri

string text = "(507) 751 45 92aksmdkamsdkmaksmdkmaskdmaksd";

Regex regex = new Regex(@"[(\d{3})]\s\d{3}\s\d{2}\s\d{2}");

Match match = regex.Match(text);

Console.WriteLine(\$"Success : {match.Success}");

Console.WriteLine(\$"Value : {match.Value}");

Console.WriteLine(\$"Index : {match.Index}");

Console.WriteLine(\$"Length : {match.Length}");

#endregion

< 1ms elapsed

C:\Users\Gençay\Desktop\duzenli_ifadeler\duzenli_ifa

Success : True

Value : (507) 751 45 92

Index : 0

Length : 15